



04770  
2100  
#4  
2131

Docket No. 017.40746X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): OSKARI, et al  
Serial No.: 09/976,091  
Filed: October 15, 2001  
Title: WIRELESS LOCK SYSTEM

RECEIVED  
DEC 27 2001  
Technology Center 2100

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of  
Patents and Trademarks  
Washington, D.C. 20231

December 18, 2001

Sir:

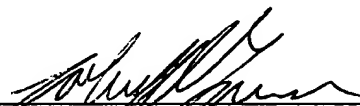
Under the provisions of 35 USC 119 and 37 CFR 1.55, the  
applicant(s) hereby claim(s) the right of priority based on:

Finnish Patent Application No. 20002255  
Filed: October 13, 2000

A certified copy of said Finnish Patent Application is  
attached.

Respectfully submitted,

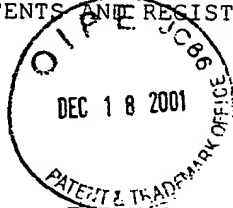
ANTONELLI, TERRY, STOUT & KRAUS, LLP

  
Robert F. Gnuse  
Registration No. 27,295

RFG/gfa  
Attachment

PATENTTI- JA REKISTERIHALLITUS  
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 3.10.2001



RECEIVED

DEC 27 2001

Technology Center 2100

ETUOIKEUSTODISTUS  
PRIORITY DOCUMENT



Hakija  
Applicant

Koskimies, Olli Oskari  
Helsinki

Patenttihakemus nro  
Patent application no

20002255

Tekemispäivä  
Filing date

13.10.2000

Kansainvälinen luokka  
International class

E05B

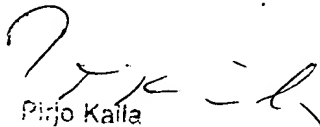
CERTIFIED COPY OF  
PRIORITY DOCUMENT

Keksinnön nimitys  
Title of invention

"A method for lock management and control"  
"Menetelmä lukkojen hallintaan ja kontrollointiin"

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.

  
Pirjo Kaila  
Tutkimussihteeri

Maksu 300,- mk  
Fee 300,- FIM

Osoite: Arkadiankatu 6 A Puhelin: 09 6939 500 Telefax: 09 6939 5328  
P.O.Box 1160 Telephone: + 358 9 6939 500 Telefax: + 358 9 6939 5328  
FIN-00101 Helsinki, FINLAND

# A Method for Lock Management and Control

## 1 Field of Invention

The invention relates to controlling and managing locks using electronic lock, key and control devices, and to creating easily distributable temporary or otherwise limited keys to said locks.

## 2 Prior Art

Current locks are all based on the principle of a shared secret between the lock and the key. There are four main lock types, and each has its problems:

1) Mechanical locks, where the secret is the way the key is formed.

- The user has to carry a separate key for each lock he can access. The keys have to be dug out of handbag or pocket every time a door is opened.
- Distribution of keys is cumbersome and has to be done by hand.
- Creating keys requires special equipment.
- Invalidating keys is hard.
- The use of keys cannot easily be limited (e.g. to office hours).

2) Electronical locks with (possibly wireless) keys, where the secret is an access code stored into both lock and key.

- While a key may have space for several codes, this is uncommon and the number of codes is limited. Thus, the user still has to carry many keys, especially as the systems are incompatible with each other. Note that if the same code is used in all locks, then the owner of any lock is able to create a key for opening all the other locks. Thus, you would have to trust the owners of all locks that you use.
- Distribution of keys is cumbersome and has to be done by hand.
- Creating new keys usually requires special equipment, and even if a key can store several codes, access to the lock is required. While access to the lock is not necessary if a single, known code for the lock is always used, this would also mean that all the created keys

share the same code and cannot be separately controlled. For instance, it would not be possible to revoke just a single key.

3) Keyless mechanical or electronical locks, where the user has to remember the code and enter it whenever access is needed.

- While the user does not have to carry keys, he has to remember all his codes, which is actually worse for many people.
- Creating new keys (codes) requires access to the lock.
- While codes can be distributed electronically, they can be used by anyone, making use of secure channels necessary.
- The code can be learned by secretly observing the user as he enters the code.

4) Keyless electronical locks, where the user's fingerprint, retinal scan or other similar feature is used for identification.

- The required scanning devices are expensive.
- Creating new "keys" requires access to the lock.
- In theory, if your information is stored on a lock, the owner of that lock can use that information to e.g. create a replica of your finger for opening all locks you have access to.

### 3. General Description of Invention

The object of the invention is to provide a method for managing and controlling locks, which solves or alleviates the problems described in the previous section, increases security and enables creation of temporary or otherwise limited, easily distributable keys (also referred to as "tickets"). This is achieved by means of features described in the patent claims.

In accordance with the invention, digital signatures and public key cryptography are used to solve the problems mentioned in the previous sections. Each user has a key device. Preferably a user has only one key device in use at a time. Key devices contain both a public and a secret key (hereafter a public key - secret key combination is referred to as an RSA key pair<sup>1</sup>). Lock devices contain the

<sup>1</sup>The term RSA is used since it is the most well known public-key cryptosystem. However, some other public key cryptosystem could also be used.

public keys of all the users that have permission to open the lock. Additionally, separate control devices may be used for controlling lock and key devices to minimize the need for control panels, allowing key and lock devices to be small.

In the preferred embodiment of the invention, wireless communication is used between lock devices, key devices and control devices. The wireless communication devices are preferably Bluetooth devices, for reasons of price, power consumption, compatibility and size. In the following, it is assumed that Bluetooth devices are used, as the described methods utilize Bluetooth security features. However, other systems that offer basic authentication and encryption support could also be used.

A user is given the right to open a lock ("given a key") by storing the public key of the user's key device on the lock. Note that in this way a key device can open an infinite amount of locks, but only needs to store one RSA key pair. Also, the owner of a lock is unable to open any other locks the key device can open, since he only knows the public key of the key device.

When a key device detects a nearby lock device, it requests access. The lock device issues a challenge in the form of a random code. The key device encrypts the code with its secret key, and sends the result to the lock, who decrypts it with the public key of the key device that was stored in the lock earlier. If the decrypted code is the same that the lock device originally sent, the lock opens.

Access permissions, or "tickets" can be created by specifying a list of limitations (such as who is able to use the permission and when), and digitally signing the permission with the secret key of a user that has access to the lock in question (meaning, his public key is stored in the lock). The lock is then able to verify that the permission was created by a user authorized to do so. Since the ticket can be limited to a certain person by including the public key of the person in question in the ticket, unsecure channels (such as email) can be used in distributing tickets. Even if someone else is able to copy the ticket, he cannot use it without knowing the secret key of the legitimate user. Tickets are stored on key devices. The number of tickets a key device can store is limited by the amount of storage space (non-volatile memory) available.

Note that while creating keys requires access to the lock device, tickets can be created just by using a key device whose public key is stored in the lock device. It is even possible to create tickets that allow the creation of more tickets. The ticket holder simply creates a new ticket, signed with his own secret key, and appends the original ticket (a more detailed description is provided in the next

section). This means that tickets are in fact equivalent to keys in terms of functionality - the only drawback is that more storage space is required in the key device.

## 4 Detailed Description of the Preferred Embodiment

The present invention relies on the use of digital signatures to authenticate tickets. It further  
5 relies on chaining signatures in such a fashion that each signature authenticates the next one, to authenticate tickets created from other tickets (delegated certificates). By way of illustration and not by way of limitation, specific ticket contents and methods for verifying and authenticating tickets and keys are given in the description.

Similarly, the listed components of a device show the preferred embodiment of the present invention,  
10 and other configurations are also possible. For example, instead of placing the confirmation input device on the key device, it can be placed on the control device, which then forwards the confirmation to the key device. For another example, a key device might have an LCD display that shows the tickets that have been stored to the device.

Similarly, lists of information that a device may contain describe the preferred embodiment of the  
15 invention. The present invention can be adapted for a variety of needs by varying the information present on devices (including, but not limited to, the variation possibilities given by the lists of optional information). For example, storing user names on key devices is useful, but not essential. Adding information to devices allows a number of specialized uses. For example, by adding a social security number and the public key of the key device, both digitally signed by a state agency, a key  
20 device can be used for identification purposes.

The reader should be familiar with Bluetooth security features (specifically the different types of link keys and their use) and public key cryptography. Information about Bluetooth is available from <http://www.bluetooth.com/>, and a good starting point for public key cryptography is the Usenet cryptography FAQ at e.g. <http://www.landfield.com/faqs/cryptography-faq/>.

25 It should be noted that whenever digital signatures by a "well-known trusted authority" are spoken of, the signatures can be chained so that there is one known central authority, who gives out authorizations to other organizations to create signatures. The authorization is then in the form of

the public key of the receiving organization, encrypted with the private key of the central authority. The organization can then sign information with its own secret key, and enclose the authorization from the central authority. The authenticity of the information can be checked by first decrypting the authorization with the (well-known) public key of the central authority (proving the organization has the right to produce signatures), using the resulting public key of the organization to decrypt the signature (proving that the signature is produced by the organization the authorization is for), and checking that the signature matches the information (proving that the information has not been tampered with). In this way, one only has to know the public key of the central authority to check the authenticity of any information, but the central authority does not itself have to sign all the information - it can delegate that to other trusted parties which do not have to be well-known.

#### 4.1 Brief Description of the Drawings

The present invention is more easily understood with reference to the drawings, in which:

**Fig. 1** is the flowchart for opening a lock with a key, from the key device point of view.

**Fig. 2** is the flowchart for opening a lock with a key, from the lock device point of view.

**Fig. 3** is the flowchart for opening a lock with a ticket, from the key device point of view.

**Fig. 4** is the flowchart for opening a lock with a ticket, from the lock device point of view.

**Fig. 5** is the flowchart for verifying a ticket, from the lock device point of view.

**Fig. 6** is the key for the different symbols in the flowcharts.

The numbering in the flowcharts follows the following conventions:

- o The first digit in a number is the number of the figure. Thus, when a number is given, a reference to the figure is not necessary.
- o Even thousands signify the whole flowchart, and are only used in flowchart references (e.g. 5000 signifies the flowchart in Fig. 5).
- o Except for iteration and flowchart references, the numbering is ordered so that if item X happens after item Y, then X has a number greater than Y.

## 4.2 Key Devices

A key device (hereafter KD) consists of a power source, a processing unit, storage (volatile and non-volatile memory), a communication device (preferably a Bluetooth wireless communication device), a confirmation input device (e.g. a button) and a confirmation request output device (e.g. a LED light). It may also have an emergency power socket that can be connected to a similarly equipped lock device (hereafter LD). A KD may further have a motion detector that allows it to switch off in order to conserve power when the KD is not moving. A KD may also have additional output devices for signaling success, failure, low power etc.

A KD stores the following information:

- A unique key device identifier, hereafter KID. A KID may or may not be changeable. Using e.g. the Bluetooth device address would make duplicating KDs impossible.
- A Unit Key (as per Bluetooth specification).
- A code used for controlling the KD, hereafter KD PIN code.
- An RSA key pair.
- The tickets of the user (maximum number may vary).
- User name.

Optionally, a KD may also store the following information:

- List of lock device identifiers for the LDs the KD can open, possibly also their human-readable names.
- Combination keys (as per Bluetooth specification) of LDs.
- User information, such as employee number, address, etc. The information may be encrypted and/or digitally signed.
- User authentication information, such as an access code, a fingerprint, a retinal scan, etc. The information may be encrypted and/or digitally signed. This can be used to guard against stolen KDs.
- A use counter for each ticket with a limited number of uses.



o An authentication token that contains the Bluetooth address (or a similar, unique network address if another communication technology is used) of the KD, digitally signed by a well-known trusted authority. The idea with the token is that such a token is only given to "secure" devices that cannot easily be used to copy tickets with a limited number of uses or to otherwise commit fraudulent acts. The token can be used to certify that the KD behaves in a certain way. Note that the token offers only limited security, since network addresses can of course be duplicated, or the Bluetooth device of a secure KD can be removed and planted in an unsecure device. However, the token scheme significantly raises the effort needed to commit fraudulent acts, since a KD cannot then be compromised using software alone. The system then compares favorably to traditional paper tickets which are easy to forge. Note also that copied tickets are a problem only if the ticket has a limited number of uses, there are several lock devices where they can be used, the lock devices cannot constantly communicate with each other to share information about ticket use, and fraud detection after the fact is not sufficient.

15 The advantage of placing user authenticating information on the KD is that the information can be used by different, independent LDs to authenticate the user. The disadvantage is that if a KD is stolen, the information could conceivably be read from it. Also, owners of other LDs would know the information since the user has to enter it when opening an LD. Thus, only information like fingerprints and retinal scans, which are known by other LDs that use similar security features in  
20 any case, should be stored unencrypted on the KD. The information should still be digitally signed, together with the KID, by some well-known trusted authority to guard against stolen KDs whose authentication information has been overwritten with counterfeited information.

Authentication information like access codes, that depend on the information being secret, should be encrypted by the LD that stores it on the KD. The information cannot then be generally used,  
25 but it is still useful, since it can be used by different LDs which are owned by the same authority but do not have contact with each other (other than sharing the same encryption key).

It should be noted that even storing unencrypted and unsigned authentication information on the KD is still a valuable security feature, since even if the KD is stolen, reading or counterfeiting the information requires technical knowledge and equipment unavailable to most criminals. For

example, a fingerprint stored on the KD in unencrypted form significantly enhances security for LDs that have fingerprint scanning capability.

### 4.3 Lock Devices

An LD consists of a power source, a processing unit, storage (volatile and non-volatile memory), a communication device (preferably a Bluetooth wireless communication device), and (assuming the LD is installed as a door lock) a device that mechanically locks and unlocks the door. An LD may also have an emergency power socket for KDs that have run out of power. An LD may further have input devices for reading user authentication information, such as keypads, fingerprint or retinal scanning devices, etc. An LD stores the following information:

- A unique lock device identifier, hereafter LID. A LID must be changeable to support the copying of locks.
- A human-readable name for the lock.
- A Confirm flag that specifies whether users should confirm unlocking the door by operating the confirmation input device on the KD.
- A code used for controlling the LD, hereafter LD PIN code.
- For each KD that can open the LD:
  - KID.
  - User name.
  - Bluetooth Link key.
  - KD public key.
  - Access rights (e.g. time period when the KD has access, whether the KD is authorized to create new keys or tickets, what kind (e.g. at most one-day) of tickets the KD can create if any, whether they KD is authorized to perform key management operations on the lock device).

Optionally, an LD may also store the following information:

- User authenticating information, such as access codes, fingerprints, retinal scans, etc., to guard against stolen KDs. Note that this information could also be stored on the KD.

- A key for encrypting and decrypting the above information when they are stored on a KD.
  - A list of untrusted public keys and ticket identifiers (hereafter blacklist, see also Creating Tickets below). Any ticket that contains one of these public keys or ticket identifiers is invalid. Also, any KD whose public key is in this list cannot store its public key on the LD.
- 5 When adding a key to this list, any KD with that public key must also be removed from the LD's KD database.

Additionally, each ticket identifier on the blacklist may have a validity date, after which the ticket is invalid in any case. This allows obsolete information to be purged from the blacklist.

Furthermore, each ticket identifier on the blacklist may have a counter that gives the number  
 10 of times the ticket has been used. This allows tickets that can be used  $n$  times.

Link keys are used by Bluetooth for authentication. Normally, Unit keys of KDs are used. This allows the KD to authenticate an LD as one of the LDs it has stored its Unit key on. Alternatively, Combination keys can be used to allow LDs to be authenticated individually. This may sometimes be useful for management operations. The disadvantage is that a separate Link key has to be stored  
 15 on the KD for each LD.

Changing the Unit key of a KD will make all locks fail authentication to the KD. However, the KD will still be authenticated to the locks since the KD's RSA pair is used for that. Thus, the KD will still be able to open LDs.

LIDs are hierarchical(e.g. "customer number" - "site number" - "lock number") to facilitate master  
 20 keys: If a KD can open an LD, it can also open any LD beneath it in the hierarchy. Technically this is only necessary for the tickets, since the KDs rely on the LDs to check whether their public key is stored on the LD or not.

Finally, note that the wireless nature of the solution allows LDs to be placed inside the door, making tampering impossible. If the LD includes an emergency power socket for out-of-power KDs, the  
 25 socket has to be located on the outside, but since it is used solely for power transfer it cannot be used for tampering with the lock. Of course, placing the LD on the inside of a door is feasible only if there is some other way of getting inside if the LD malfunctions.

#### 4.4 Control Devices

KDs and LDs are controlled via separate control device (hereafter CD), that also includes a Bluetooth device. LDs can also have a built-in CD, or a wireline connection to an external control system.

- 5 A CD is also a KD. The access rights for the CD's public key stored on the lock must enable control operations.

To create initial keys, an LD PIN is used (as per Bluetooth specification) for both authentication and encryption. The KD PIN is used for authentication and encryption between KDs and CDs.

If the Bluetooth technology is successful, many mobile phones will be equipped with Bluetooth  
10 devices to enable them to be connected to similarly equipped computers. A Bluetooth-enabled mobile phone is also an optimal CD:

- o Most people will have one.
- o PIN-based KD security controls (e.g. enable/disable KD) can be tied to those of the phone (e.g. enable/disable outgoing calls).
- 15 o Keys and tickets can be transmitted with the phone.
- o A phone can also itself function as a KD. This is extremely valuable, since it would make achieving "critical mass" for the system much easier.

#### 4.5 Creating Keys

A key can be created whenever the LD and the KD are in contact. A CD must be used to activate  
20 the LD's key creation sequence. The LD will then show (via the controller) the user names of all unknown key devices in range. A key device is selected by the user of the LD.

Optionally, a temporary PIN code can be selected for authentication and encryption between the LD and the KD, as per Bluetooth specification. In that case, the PIN must also be entered to the KD using a controller.

- 25 The LD sends a key registration request to the KD. If a temporary PIN was not used, the KD signals the user of the KD via the confirmation request output device, and awaits an action on the

confirmation input device. After the user has activated the input device, the KD sends its KID, user name, link key and public key to the LD. Access rights for the KD must then be entered to the LD via its CD.

The link key is either the KD's Unit key, or a combined key can be created (as per Bluetooth specification). In the latter case, both the combined key and the LID must be stored on the KD. In any case, the KD may store the LID to keep track of the locks it has access to.

KD registration can be done remotely by sending the above information via any electronic media to the controller. While the media need not be secure against eavesdropping, it should be secure against an attacker replacing the information with his own.

- 10 A KD that is not a CD can also have the right to create new keys. In that case, a CD must be used to ask the KD to create the key and for controlling the process. The KD will effectively act as a mediator between the CD and the LD.

## 4.6 Using Keys

Figure 1 shows the process of using a key, from the KD point of view, as a flowchart. Figure 2 shows the process of using a key, from the LD point of view, as a flowchart.

- 1) The LD broadcasts its service (2020, 1020).
- 2) The KD sends its KID (1030, 2030) to the LD, which looks up the KID in its database (2040). The LD then replies with its LID, Confirm flag and a flag that tells if the lock knows the KID (true in this case) (2050, 1040). If the confirm flag was true (1060), the KD signals a confirmation request to the user (1070) and awaits confirmation.
- 3) The KD authenticates the lock with its link key according to Bluetooth specification (KD challenges, LD responds) (1090, 2070, 2080, 2090, 1100). Note that the LD finds the KD's link key based on the KID, not on the KD's Bluetooth address.
- 4) Encryption based on the link key is taken into use (1120, 2100). The LD sends a random number to the KD (2110, 1130). The KD encodes the random number and its Bluetooth address with its secret key (1140), and sends the result to the LD (1150, 2120).

5) The LD decrypts the encrypted number and address using the public key that corresponds to the KID (2130). If the decryption is successful (meaning that the key is correct)(2140), the address matches the one the KD has (2150), and the number matches the one the LD sent (2160), and the access rights of the user allow him to get in now (2170), then the user is authorized to open the lock(2190, 1160–1180, 2200). Closing the lock again could be based on a simple timeout (lock stays open for a predefined time interval), but preferably the LD would use radio signal strength to get some measure of the distance to the KD, and when signal strength is sufficiently weak (2210), close the lock again(2220).

Preferably, public key cryptography is used to authenticate KDs to LDs, because then it is enough to store a single RSA key pair on the KD. However, it is also possible to store a separate key for each LD on the KD. This has the disadvantage that more storage is required, but the advantage that a more efficient cryptographic method (such as block cipher) can be used instead of public key cryptography. Preferably, Bluetooth combination keys should be used, since they can then easily also be used to individually authenticate LDs.

#### 4.7 Creating Tickets

A KD contains all the tickets of its user. A ticket can be created based either on a key or another ticket. In the former case we use the term *original ticket*, and in the latter case we use the term *derivative ticket*. In literature, the term *delegated certificate* is also used for the same concept.

Each ticket is assigned a unique ticket identifier (hereafter TID) when creating it. A TID is useful for two things: It allows tickets to be revoked individually (by storing the TID on a blacklist on the LD) and it allows single-use tickets (by marking the ticket as single-use in its access limits and having the LD store the TID of tickets so marked on its blacklist after use). The method can be expanded to tickets with  $n$  uses (hereafter  $n$ -use or N-use tickets) by storing the total number of allowed uses in the ticket's access limits, and by adding a count to the blacklist that is increased with each access. The ticket is then refused only when the access would actually increase the count in the blacklist above the total number of allowed accesses stored in the ticket's access limits. Note that tickets with a limited number of uses should preferably also have a validity time limit (a not-after date) to allow LD's to purge obsolete information from their blacklists.

If a  $n$ -use ticket can be used for several LDs, which are not in contact with each other, it is possible to copy the ticket and use it several times, since the LDs cannot keep track of the total number of uses. If LDs store ticket use information, this can later be detected by combining the data in the different LDs, and the fraudulent user can then be identified. However, detection after the fact is obviously not appealing. The security can be strengthened by requiring that the KDs of the ticket receivers have an authentication token from a well-known trusted authority, that contains the KD's Bluetooth address (or a similar unique network address if another communication technology is used), digitally signed by the authority. The idea is that such tokens are only given to KDs that cannot easily be used for copying tickets. Such KDs would of course also only allow derivative tickets to be created for KDs that have a similar authentication token.

Preferrably, a CD is used for creating tickets. If the LD supports a blacklist of TIDs, the TID (that is included in the ticket) should preferrably be stored on the CD, so that, if necessary, the ticket can later be revoked by adding the TID to the blacklist. If the ticket has a validity time limit, that should be stored on the CD as well, so that it also can be stored on the blacklist.

An ticket based on a key contains a LID, a TID, the KID of the granter, a link key, the public key of the receiver, access limits, the maximum number of levels of derivative tickets, limits on the derivative tickets (e.g. only one-day derivative tickets may be created) and a checksum encoded with the secret key of a user authorized to grant such tickets.

The link key of a ticket is created using a (well-known) secure one-way hash on the granter's link key and the LID.

A derivative ticket contains a LID, a TID, the public key of the receiver, access limits, the maximum number of levels of derivative tickets, limits on the derivative tickets, a checksum encoded with the secret key of the granter, and additionally the original ticket that the granter has. Note that derivative tickets may be nested to an arbitrary depth.

A KD may store and maintain a use count for a  $n$ -use ticket to keep track of how many uses the ticket has left. The use count is then increased with each use, and compared to the total number of uses in the ticket to get the number of remaining uses (alternatively, a decreasing counter could also be used). Note that when a  $n$ -use derivative ticket is used, the use count is increased for all its  $n$ -use nested tickets as well. This means that when a  $n$ -use ticket is used to create derivative tickets,

the owner of the first ticket cannot know any more how many uses his ticket has at a certain time. We call tickets with this kind of shared number of uses *shared* tickets. Alternatively, if the derived ticket is created with fewer uses than the first ticket, and the number of uses of the derived ticket is added to the use count of the first ticket, the owners of both tickets then know exactly how many uses their tickets are good for. We call tickets with this kind of individual number of uses *unshared* tickets. KDs should maintain information about the sharedness of their tickets. Note that keeping a use count for shared tickets only gives you the maximum number of remaining uses. Also note that if a ticket is used to create a shared derivative ticket, the first ticket automatically becomes shared as well.

- 10 If public key cryptography is not used, then the checksum in an original ticket T1 is encrypted with the secret key K1 that the KD creating the ticket has for the lock. Additionally, a new secret key K2 (preferably a Combined key as per Bluetooth specification) is created and shared by the KD creating the ticket (KD1) and the KD receiving the ticket (KD2). Instead of the public key of the receiver, KD1 includes in the ticket K2, encrypted with K1. KD2 stores the K2 together with the ticket. The LD can then verify the ticket by decrypting the checksum with the secret key it has stored for the KD1 (K1), and verifying that the checksum is correct. The LD can further verify that KD2 is the KD the ticket was created for, by similarly decrypting K2 with K1, and issuing an authentication challenge as per Bluetooth specification to KD2, using K2 as the link key.

- If public key cryptography is not used, then derivative tickets are created as follows: Assume KD2 from previous paragraph wishes to create a derivative ticket T2 for another KD (KD3). It proceeds otherwise as with public key cryptography, but encrypts the checksum using the secret key K2 it has stored for its own ticket. A new secret key K3 is then created and shared by KD2 and KD3 (preferably a Combined key as per Bluetooth specification). KD2 encrypts K3 with K2, and includes it in the new ticket instead of the public key of the receiver. When the LD verifies the ticket, it can obtain K2 from the original (nested) ticket as explained in the previous paragraph. It can then decrypt K3, and authenticate KD3 as per Bluetooth specification, using K3 as the link key. In this way, tickets can be nested arbitrarily deep.

Below is a description of tickets in Backus-Naur Form notation. It assumes public key cryptography is used. Note that on derivative tickets, the term "granter" refers to the immediate granter (who



has a ticket), not the original one (who has a key).

$\langle \text{Ticket} \rangle := \langle \text{Original\_ticket} \rangle \mid \langle \text{Derivative\_ticket} \rangle$

5  $\langle \text{Original\_ticket} \rangle := \langle \text{LID} \rangle \langle \text{TID} \rangle \langle \text{Granter\_KID} \rangle \langle \text{Link\_key} \rangle$   
 $\langle \text{Receiver\_public\_key} \rangle \langle \text{Access\_limits} \rangle$   
 $\langle \text{Max\_derivation\_level} \rangle \langle \text{Derivation\_limits} \rangle$   
 $\langle \text{Checksum} \rangle$

10  $\langle \text{LID} \rangle := \langle \text{LID} \rangle$  of the LD the ticket can open.

$\langle \text{TID} \rangle :=$  Unique ticket identifier.

$\langle \text{Granter\_KID} \rangle :=$  KID of the granter's KD.

15

$\langle \text{Link\_key} \rangle :=$  A link key created by applying a well-known secure one-way hash function (e.g. MD5 or SH-1) on the  $\langle \text{LID} \rangle$  and the link key that is stored on this LD for the granter's KD.

20  $\langle \text{Receiver\_public\_key} \rangle :=$  The public key of the KD of the receiver of the ticket.

$\langle \text{Access\_limits} \rangle :=$  Limitations on when the ticket is valid. For example, a time period, or a counter that says how many times this ticket can be used.

25

<Max\_derivation\_level> := How many levels of derivative tickets can be  
 created. For example: A <Max\_derivation\_level>  
 of zero forbids derivative tickets entirely. A  
 level of two allows the receiver of a  
 derivative ticket, that was made by the  
 receiver of an original ticket, still create  
 derivative tickets, but those tickets cannot  
 be used to make derivative tickets. A special  
 value such as -1 can be used to allow infinite  
 levels of derivative tickets.

<Derivation\_limits> := Limits on what kind of derivative tickets may be  
 created. For example, a ticket may be limited so  
 that it only allows creation of one-day  
 derivative tickets.

<Checksum> := A secure checksum (e.g. MD5) of all the other information  
 in the ticket (excluding nested tickets), encrypted with  
 the secret key of the granter's KD.

<Derivative\_ticket> := <LID> <TID> <Receiver\_public\_key> <Access\_limits>  
 <Max\_derivation\_level> <Derivation\_limits>  
 <Checksum> <Ticket>

## 4.8 Using Tickets

Figures 1 and 3 show the process of using a ticket, from the KD point of view, as a flowchart.  
 Figures 2, 4 and 5 show the process of using a ticket, from the LD point of view, as a flowchart.

- 1) The LD broadcasts its service (2020,1020).

- 2) The KD sends its KID (1030,2030) and the LD replies with its LID, Confirm flag and a flag that tells if the LD knows the KID (false in this case) (2040, 2050, 1040, 1050).
- 3) The KD finds that it has a valid ticket for the LD (1200). If the confirm flag was true (1210), the KD signals a confirmation request to the user (1220) and awaits confirmation.
- 4) The KD tells the lock the KID of the granter of the (if nested, innermost) ticket (3030). It authenticates the LD with the ticket link key according to Bluetooth specification (KD challenges, LD responds) (3030, 4030–4080, 3040, 3050) . The LD finds the granter's link key based on the granter KID (4040), and can then create the required link key by applying the hash function (4050).
- 5) Encryption based on the link key is taken into use (3060, 4090). The LD sends a random number to the KD (4100, 3070).
- 6) The KD encodes the random number and its Bluetooth address with its secret key (3080) and adds the ticket. The result is sent to the LD (3090, 4110).
- 7) The LD decrypts the encrypted number and address using the receiver public key in the ticket (4120).
- 8) If the decryption succeeds (meaning the key was correct) (4130), the decrypted address matches the KD's address (4140), the decrypted number matches the previously sent random number (4150), the ticket is successfully verified (5000, 4160), the access rights of the user allow him to get in now (4170), and no TID or public key in the ticket is blacklisted (or, in the case of  $n$ -use tickets, above their use limit)(4180), then the user is authorized to open the lock(4190, 3100, 3110, 3120, 3140, 4200). If the ticket is  $n$ -use, and unshared, the KD may increase its use count to keep track of how many uses are left (3115). Also, the LD may include in the Access Granted message (4190) the number of times the ticket has been used. If the ticket is shared, this allows the KD to update its use counter to reflect the current situation.
- 9) Closing the lock again could be based on a simple timeout (lock stays open for a predefined time interval), but preferably the LD would use radio signal strength to get some measure of the distance to the KD, and when signal strength is sufficiently weak (4210), close the lock again(4220).

- 10) If the ticket's access limits specify that it is a  $n$ -use ticket (4230), then all TIDs in the ticket with a limited number of uses (nested tickets are included) are stored on the blacklist with a count of 1. If they have a validity limit date, that is stored on the blacklist as well, to allow it later to be purged of obsolete information. If some were already blacklisted, then the count for those is increased in the blacklist (4240).

## 4.9 Verifying Tickets

Figure 5 shows the verification process of a ticket.

- 1) If the ticket is not derivative (5020), the LD decrypts the checksum (5040) with the public key that corresponds to granter KID (5030), and verifies that the decryption succeeded (meaning that key was correct) (5050), the checksum is correct (5060, 5070), and that the granter KID is authorized to grant the ticket (access rights are stored within the LD) (5080, 5090, 5100).
- 2) If the ticket is derivative (5020), the LD decrypts the checksum with the public key of the granter (5120), which is the receiver public key in the original (nested) ticket (5110), and verifies that the decryption succeeded (meaning that key was correct) (5130), and the checksum is correct (5140, 5150). It then recursively verifies the original ticket (5000, 5160), and checks that it allowed at least one level of derivative tickets (5170). It also checks that the level of derivative tickets allowed increases by at least one at each recursion level (5180), that limits on derivative tickets are observed (5190), that derivation limits are included in the derivation limits of the nested ticket (5200), and that access limits are always included in the access limits of the nested ticket (5210). The last two inclusion checks are necessary to prevent people from creating derivative tickets that are less restrictive than the ticket they possess.
- 3) If the ticket passes all the checks, it has been successfully verified (5220).

## 4.10 Key Management in Locks

Keys can be added, removed, and their access rights can be modified. These are simple database operations. Keys are structured as a forest (a group of tree hierarchies). Each key must have been authorized either by an LD PIN (root keys), or by another key. The parent of a key is the key that authorized it.

Keys can be removed or their access rights modified only by keys above them in the hierarchy. Possibly a KD can also remove itself from a lock. PIN authorization allows everything ("root access"). A key cannot have wider access rights than its parent.

If a key is removed because it has been compromised, all keys below it in the hierarchy should also  
 5 be removed.

#### 4.11 Changing Keys

For security reasons, users may want to change their secret keys periodically. Update of LDs can be done automatically, by allowing public keys in LDs to be updated by the owner of the keys. The KD must then store both the old and the new RSA key pair until all LDs have been updated.

- 10 The KD can remove the old RSA key pair once all LDs have been updated. This requires that the KD stores the LIDs of all LDs it has access to. As this increases the memory requirements of the KD, it would be an optional feature. The other option is that the old RSA key pair is removed manually by the user.

#### 4.12 Practical Issues

- 15 It is often not desirable for a door to unlock because someone with a KD walks nearby on the inside. The lock should be shielded against radio signals from that direction (doors can be opened from the inside without unlocking).

However, if the door should block movement also from the inside, the radio signals should be restricted so that there is a very small area immediately next to the door where KDs will open the  
 20 door.

Also, to prevent unauthorized people from sneaking in when someone with a KD (that allows access) walks *past* the door, either the locations where KDs are effective on the outside should be limited to the immediate vicinity of the door, or confirmation should be required. Still another possibility is that an LD only opens the door if the KD stays close (determined by signal strength) for some  
 25 time.

### 4.13 Applications

- o Need to search for keys is removed. Doors unlock automatically (unless confirmation is required).
- o There is no key chain weighing down a user's pockets, nor a need to remember which key is which. One key opens all doors (as long as they have compatible locks).
- o There is no need to have a key hidden somewhere. For example, if a user is already on the road when he realizes someone should water his plants, he can send a ticket to his friend that is valid until the end of his trip. If he sets the ticket's max derivation level to higher than zero, his friend can in his turn delegate the responsibility to someone else and create a derivative ticket for that person.
- o If a user is stuck in traffic, and his friends are coming to visit, he can send them a one-time ticket.
- o Moving is simplified. The key data and Lock ID from the old lock are copied to the controller, the new lock is cleared, and the data from the old lock is copied to it.
- o Keys for friends and relatives can be created instantly and for free, and their access can be limited to reasonable hours.
- o A temporary ticket can be sent via Internet to allow an Internet store to deliver goods when you are not at home. The ticket might open your home door, but more likely a separate delivery area. Since the ticket is temporary (and/or one-time), it cannot be used by the store employee later to open the lock when inside would be deliveries from another store. If there is a separate delivery area, there is no need to trust the store.
- o If a car's ignition lock is replaced with an LD, car sellers can give limited-time tickets when you take a car for a test drive.
- o By installing LDs that control access to movie theatres, busses, etc., electronic tickets in KDs can be used instead of traditional paper tickets. Of course, KD public keys could be stored on the LDs as well, but tickets are easier to create.

## 5 Claims

What is claimed is:

1) A method for controlling and managing locks, characterized in that:

- o A public and a secret key is stored on each key device.
- o An lock device stores the public keys of the key devices that can open it, and uses them to authenticate a key device.
- o A key device can create tickets (authorizations) by:
  - i) Digitally signing them with its secret key, and
  - ii) including the public key of the receiving key device in a ticket.

Tickets are stored on the receiving key device, allowing it to be authenticated by the lock device the ticket is for.

2) A method for controlling and managing locks, characterized in that:

- o When a key device is allowed access ("given a key") to a lock device, a new secret key is generated and stored both on the key device and on the lock device. Thus, a key device has a separate secret key for each lock device it can open.
- o A lock device uses the secret key it has stored for a key device to authenticate that key device. A key device may also authenticate the lock device similarly.
- o A key device can create tickets (authorizations) by:
  - i) Digitally signing the ticket with the secret key K1 for the lock the ticket is for, and
  - ii) including in the ticket another secret key K2, which is known by the receiving key device, encrypted with K1.

The tickets are stored on the receiving key device together with K2, allowing it to be authenticated by the lock device the ticket is for.

3) A method according to claim 1, characterized in that a key device with a ticket can, if the ticket allows it, create more tickets (derivative tickets) by:

- i) Creating a new ticket that includes the public key of the receiving key device,
- ii) digitally signing it with its own secret key, and
- iii) appending the original ticket to the new ticket.

The new ticket then includes the original ticket.

4) A method according to claim 2, characterized in that a key device with a ticket T1 can, if the ticket allows it, create more tickets (derivative tickets) by:

i) Creating a new ticket T2,

ii) digitally signing it with the key K2 the key device has associated with T1,

5 iii) including a secret key K3, which is known by the receiving key device and is encrypted with K2, and

iv) appending the original ticket to the new ticket.

The new ticket then includes the original ticket.

10 5) A method according to claim 1 or claim 2, characterized in that a Bluetooth wireless communication device is used for communication, allowing wide physical compatibility between devices.

6) A method according to claim 1 or claim 2, characterized in that user authenticating information (such as an access code, a fingerprint, a retinal scan etc.) is stored on the key device.

15 7) A method according to claim 6, characterized in that the user authenticating information is encrypted and/or digitally signed by either by a lock device or by a well-known trusted authority.

8) A method according to claim 1 or claim 2, characterized in that tickets are transferred to other key devices via any electronic means, such as direct radio link, cellular data connection, SMS message or internet.

20 9) A method according to claim 1 or claim 2, characterized in that a mobile phone is used either as a key device or as control device for a key device.

25 10) A method according to claim 1 or claim 2, characterized in that the number of times a ticket can be used is limited by storing the total number of allowed uses in the ticket's access limit information, and by storing the ticket's ticket identifier, possibly together with a use count and/or a validity end date, on a another list (a so-called blacklist) on the lock device after first use. On subsequent uses, the use count is increased. The blacklist is used to check if an offered ticket has been used too many times.

11) A method according to claim 1 or claim 2, characterized in that tampering-resistant key devices have authentication tokens, digitally signed by a well-known trusted authority, that



contain information (such as the key device's Bluetooth network address) that can be used to at least some degree verify that the key device is the one the token was originally given to.

12) A method according to claim 1 or claim 2, characterized in that key device information (a key) is stored on a lock device hierarchically, so that:

- 5       o Each key must have been authorized either by a lock device PIN (root keys), or by another key.
- o The parent of a key is the key that authorized it.
- o The access rights of a key can be modified only by keys above it in the hierarchy. A key can be removed only by keys above it in the hierarchy, and possibly by itself.
- 10       o A key cannot have wider access rights than its parent.

## Abstract

A method for managing and controlling locks, that provides increased security and makes creating and using keys easier than with traditional methods. The method further enables creation of temporary or otherwise limited, easily distributable keys (also referred to as "tickets").

5 Each user has a key device. Key devices contain both a public and a secret key. Lock devices contain the public keys of all the users that have permission to open the lock, and use them to authenticate key devices that request access.

Preferably, wireless communication is used between devices. Preferably Bluetooth wireless communication is used, for reasons of price, power consumption, compatibility and size.

10 Access permissions or "tickets" can be created by specifying a list of limitations (such as who is able to use the permission and when), and digitally signing the permission with the secret key of a user that has access to the lock in question. Tickets are stored on key devices.

Fig. 1

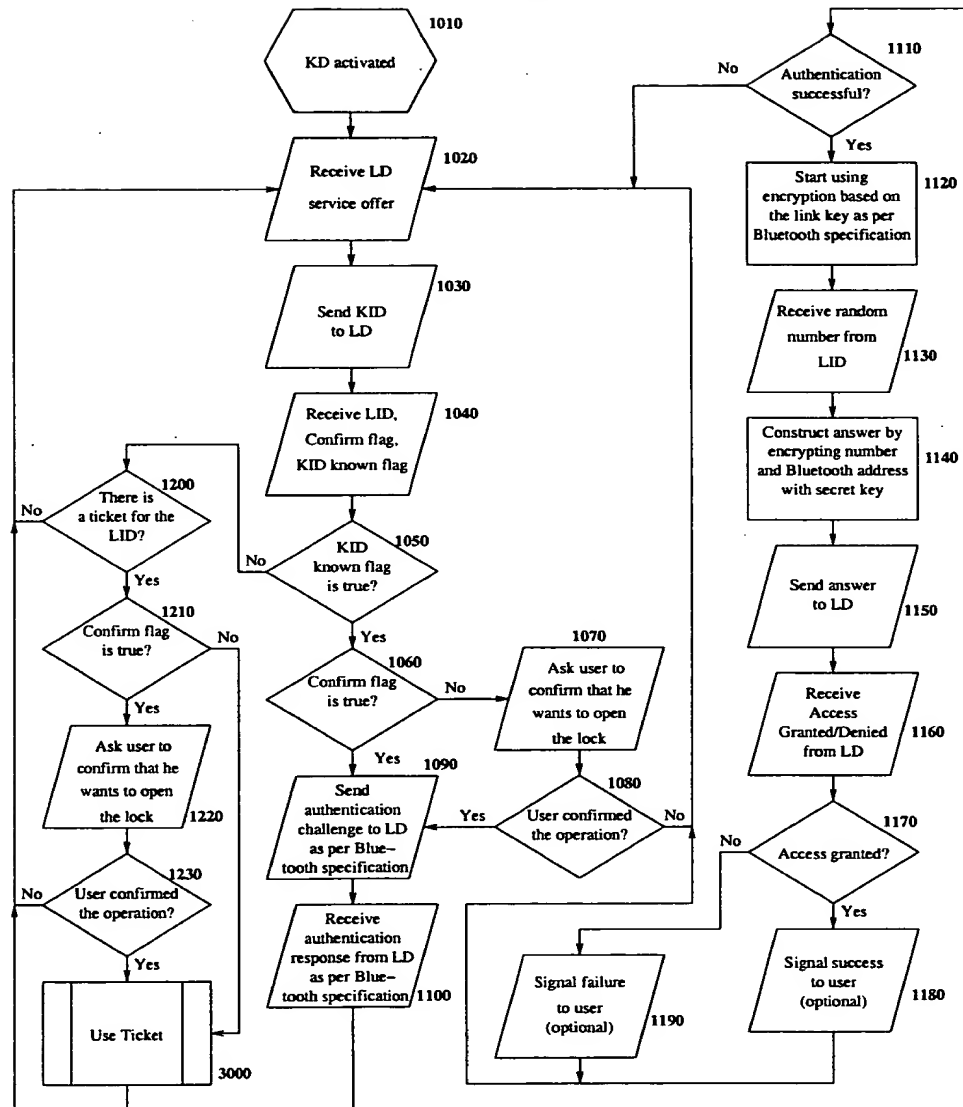


Fig. 2

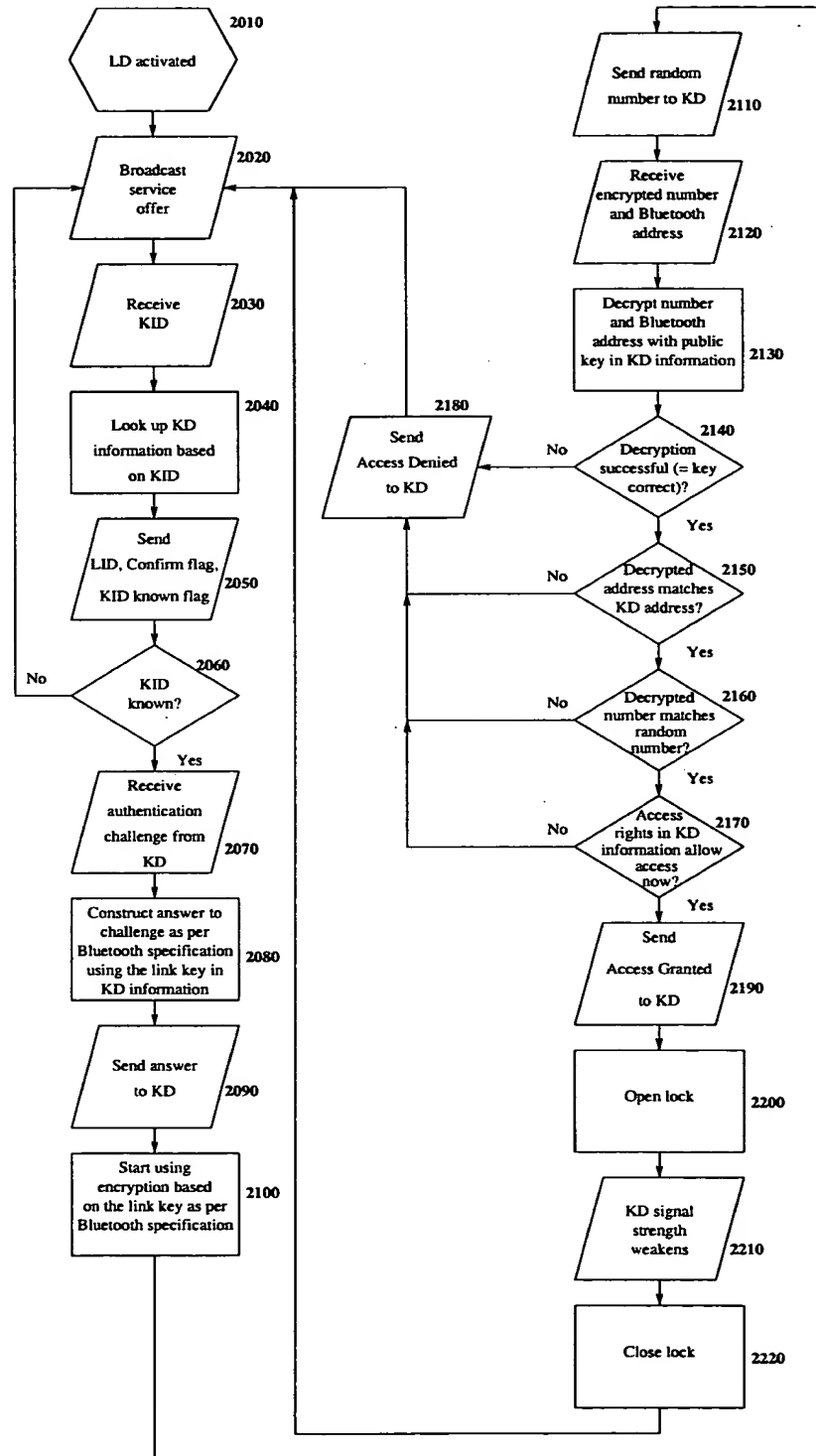


Fig. 3

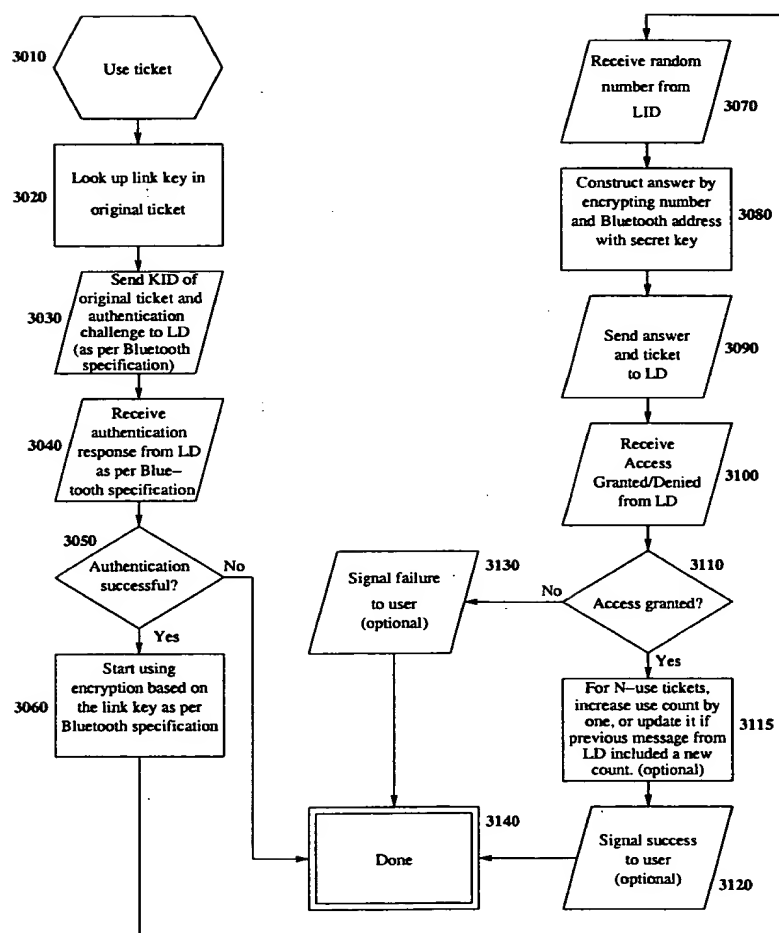
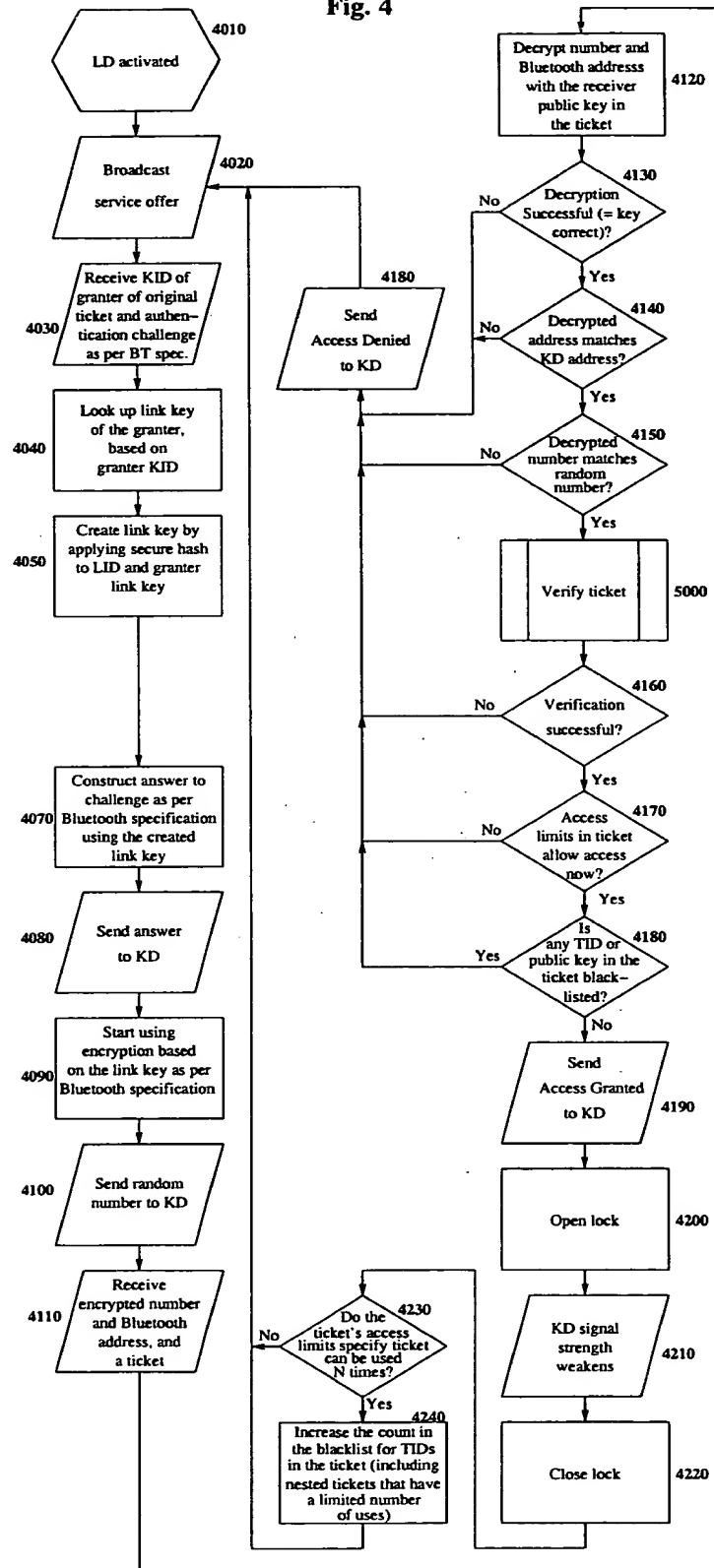
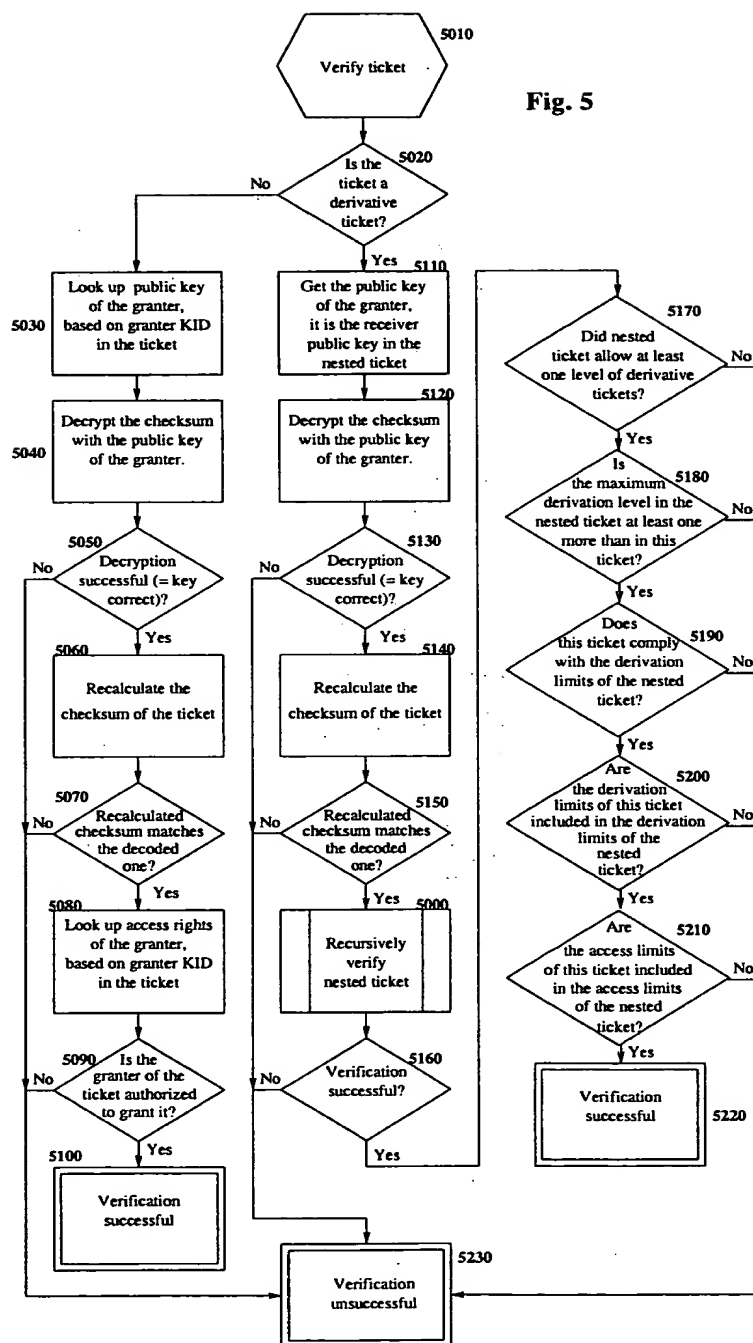


Fig. 4





**Fig. 6**